김의현

프론트엔드 개발자

연락처 +82 01083108853
이메일 rladml91gus@naver.com

사용자 경험과 협업의 힘을 믿는 3년차 프론트엔드 개발자 김의현입니다.

저는 복잡한 기능도 사용자가 쉽게 다가갈 수 있도록 만드는 데 집중합니다.
개발 초기부터 사용자 흐름을 고민하며, 기능을 단순히 구현하는 것을 넘어 '어떻게 하면 더 쉽게 사용할 수 있을까'를 항상 고민해왔습니다.

3년간의 실무에서 No Code 플랫폼, ERD 툴 등 기술 난이도가 높은 프로젝트를 경험하며, React 기반 UI 구성과 MobX 상태관리, 그리고 타입 기반 안정성 확보를 통한 구조적 개선까지 다양한 실무 과제를 해결해왔습니다.
특히 팀원들과 함께 더 나은 설계를 위해 논의하고, 유지보수성 높은 시스템을 구현하는 과정에서 협업의 중요성을 실감했습니다.

앞으로도 저는 기술을 넘어 사용자와 팀을 위한 고민을 계속하며, '사용자 중심의 기술'을 실현하는 개발자가 되겠습니다.

기술 스택

JavaScript, TypeScript, react.js, MobX, HTML/CSS, Jest, Java, Oracle, Git, gitlab, Figma, Slack

---

경력

(주)티맥스클라우드

연구원
2022.09. ~ 2024.11. (2년 3개월)

No Code / Low Code 백엔드 개발 스튜디오

프로젝트 설명:

비즈니스 요구 사항에 맞게 백엔드 서비스를 설계하고 Java 기반의 서비스 코드를 자동으로 생성할 수 있는 No Code / Low Code 백엔드 개발 스튜디오입니다.

수행 역할:

1. No Code Select 서비스 생성 기능 개발

이슈

비개발자가 SQL 지식 없이 데이터 조회 API를 생성하는 과정이 복잡하고 어려워, 쉽고 효율적인 API 생성 파이프라인 구축이 필요했습니다.

해결

사용자 친화적인 인터페이스를 위해 Drag & Drop 방식으로 테이블을 선택하고, 자동으로 상속/참조 관계를 분석하여 유효한 조합만 선택 가능하도록 UX를 설계했습니다.

선택된 테이블 컬럼 기반으로 OutDTO를 자동 매핑하는 시스템을 구현하고, 조건 및 부가 설정 시 컬럼명/연산자 자동 제안 기능을 추가했습니다.

또한, 자체 개발한 SQL Generator를 통해 복잡한 연산, 조인, 조건 등을 포함한 동적 쿼리를 생성하고, FreeMarker 템플릿 엔진을 활용하여 Java 코드를 자동 생성하도록 구현했습니다.

성과

평균 10분 이내에 Select API를 손쉽게 생성할 수 있는 기능을 제공했습니다.

---

2. 리스트 순서 변경 UX 개선

이슈

기존의 리스트 순서 변경 방식은 버튼을 눌러 이동하는 형태였고, 직관성이 떨어져 사용자 불만(VOC)이 지속적으로 접수되었습니다.

해결

React DnD 라이브러리를 도입하여 리스트 항목을 Drag & Drop 방식으로 변경할 수 있도록 개선했습니다.

useDrag, useDrop 훅을 활용해 시각적 피드백을 주었고, CSS트랜지션과 GPU 가속을 적용하여 부드러운 이동 효과를 구현했습니다.

또한, React.memo, useCallback을 적극활용해 불필요한 렌더링을 줄이고 성능 최적화를 달성했습니다.

성과

순서를 바꾸는 인터랙션이 훨씬 자연스러워졌으며, VOC가 눈에 띄게 줄고 내부 만족도 설문에서 90% 이상의 긍정적 피드백을 받았습니다.

---

3. 중앙 집중식 다이얼로그 관리 시스템 구현

이슈

프로젝트가 확장되며 다양한 다이얼로그가 여러 컴포넌트에 분산되어 관리되었고, 타입 오류 및 유지보수 이슈가 자주 발생했습니다.

해결

다이얼로그 종류를 as const 배열과 typeof를 활용해 타입으로 정의하고, 각각의 다이얼로그 Props를 개별 인터페이스로 분리하여 공통 인터페이스에서 확장 가능하도록 설계했습니다.

MobX를 통해 현재 띄울 다이얼로그 타입 및 Props를 전역에서 관리하며, 조건부 렌더링을 통해 한 곳에서 모든 다이얼로그를 통합 관리할 수 있도록 구성했습니다.

성과

20개 이상의 다이얼로그를 하나의 컴포넌트에서 효율적으로 관리할 수 있게 되었고, 타입 기반 컴파일 검증을 통해 런타임 오류를 대폭 줄였습니다.

유지보수 편의성과 코드 일관성이 크게 향상되었습니다.

---

4. 서비스 등록 및 메시지 핸들링 시스템 개선

이슈

기존 시스템은 서비스 등록 및 메시지 핸들링 로직이 분산되어 있어, 신규 서비스 추가 시마다 수동 등록이 필요하고, 타입 오류는 런타임에서 확인 가능한 구조로 디버깅을 하는데 오랜 시간이 소요되었습니다.

해결

기존 Map 기반 양방향 매핑 구조를 개선하여, 도메인별 서비스를 객체 기반으로 관리할 수 있는 구조로 설계했습니다.

Object.keys 기반의 자동 매핑 구조를 설계하여, 핸들러 등록 과정을 코드 1줄로 단순화했습니다.

핸들러 누락 시 컴파일 타임 오류가 발생하도록 강제하는 유틸리티를 구현하고, TypeScript의 keyof, 제네릭, 조건부 타입 등을 조합해 VS Code Quick Fix가 지원되는 타입 체계를 구성했습니다.

성과

컴파일 타임 오류 방지로 디버깅 시간 90% 이상 단축되었고, 신규 서비스 추가 작업 시간을 대폭 줄였습니다.

타입 기반 설계를 통해 코드의 안정성과 확장성 모두 향상되었습니다.

---

5. React 순환 의존성 문제 해결

이슈

컴포넌트 구조가 복잡해지면서 순환 의존성 문제가 반복적으로 발생했고, 이로 인해 빌드 및 런타임 오류가 자주 발생했습니다.

컴포넌트 간 직접적인 import 구조는 코드 가독성을 저하시키고, 테스트 코드를 작성하는데 어려움을 주었습니다.

해결

컴포넌트 간 직접 import를 제거하고, 상위 컴포넌트에서 Props를 통해 필요한 의존성을 주입하는 구조로 전환하여 결합도를 낮췄습니다.

타입, 유틸리티, 상수 등의 공통 요소를 별도 모듈로 분리하고 필요한 위치에서 import하여 사용할 수 있도록 설계했습니다.

성과

순환 참조 문제를 구조적으로 해결하여 빌드 및 런타임 오류를 제거했고, 이로 인해 코드 안전성이 크게 향상되었습니다.

컴포넌트 간 역할이 명확해져 코드 가독성이 향상되었고, 테스트 코드 작성이 쉬워졌습니다.

기술 스택:

- 언어 : JavaScript(ES6+), TypeScript, Java
- 프레임워크 및 라이브러리 : ReactJS, FreeMarker
- 상태 관리 : MobX
- 데이터베이스 : Tibero(Oracle)
- 협업 도구 : Figma, Slack, Google Sheet

티맥스가이아

연구원
2022.02. ~ 2022.09. (8개월)

RDBMS 기반의 데이터 모델링 설계 플랫폼

프로젝트 설명:

데이터베이스 테이블과 컬럼을 설계하고 이를 기반으로 데이터베이스 구조를 시각적으로 확인할 수 있는 데이터 모델링 설계 플랫폼입니다.

수행 역할:

1. 공통 컴포넌트 개발

이슈

팀 내 개발 효율성이 낮아지고 UI 일관성이 부족하여, 재사용 가능한 UI 컴포넌트의 공통화가 필요했습니다.

해결

Figma를 통해 디자이너와 협업하여 다양한 화면에서 공통적으로 사용되는 UI를 재사용 가능한 공통 컴포넌트로 추상화하여 구현했습니다.

Props 기반의 유연한 동작 정의 및 스타일 커스터마이징 기능을 제공하고, 기본 스타일과 사용자 정의 스타일을 병합하는 전략을 도입하여 확장성을 확보했습니다.

성과

UI 일관성을 확보하여 사용자 경험(UX)이 개선되었고, 공통 컴포넌트 사용으로 팀 내 생산성 및 유지보수성이 향상되었습니다.

---

2. 테스트 코드 도입

이슈

코드 품질 향상 및 버그 사전 방지를 위해 테스트 환경 구축의 필요성을 느꼈습니다.

해결

Jest와 React Testing Library를 활용하여 단위 테스트 환경을 구축하고 테스트 코드를 작성했습니다.

pre-push 훅을 설정하여 Git push 시 자동으로 테스트가 실행되도록 환경을 구성했습니다.

성과

테스트 환경 도입을 통해 코드 품질 및 안정성이 향상되었고, TDD(Test Driven Development)의 중요성을 인식하여 개발 방식에 점진적으로 반영하게 되었습니다.

---

3. ERD 툴 편집 모드 기능 개발

이슈

기존 생성/읽기 전용인 ERD 툴에 협업 환경을 위한 실시간 테이블 편집 기능이 필요했습니다.

해결

테이블 생성 및 편집 기능을 하나의 컴포넌트로 통합하여 UX 일관성을 확보하고 컴포넌트를 리팩토링했습니다.

React Flow 상에서 노드 더블 클릭을 통해 편집 모드로 진입할 수 있도록 구현하여 사용자 편의성을 강화했고, 편집 중 원본 데이터의 유지를 위해 Model 계층에 Deep Copy 메서드를 구현하여 데이터 무결성을 보장했습니다.

성과

실시간 편집 기능을 통해 협업 환경이 구축되었고, 생성/편집 모드 통합 및 UI 재사용으로 일관된 사용자 경험(UX)를 제공했습니다.

---

4. 데이터 마이그레이션 및 플랫폼 통합 기능 개발

이슈

Tibero 데이터베이스의 데이터를 자사 플랫폼 스키마에 맞게 마이그레이션하는 기능이 필요했고, 마이그레이션 이후 플랫폼 내에서 원활하게 개발을 이어갈 수 있도록 플랫폼 통합 기능 개발이 요구되었습니다.

해결

ETL 프로세스를 구축하여 JDBC URL, 계정 정보를 통해 DB에 연결하고 주요 메타데이터를 추출했습니다.

추출된 데이터를 플랫폼 DB 스키마에 맞게 변환하여 플랫폼 내부 테이블에 저장했습니다. 또한, ERD 시각화에 필요한 위치 정보까지 포함하여 저장하도록 구현했습니다.

성과

마이그레이션 기능을 통해 데이터 이전 시간을 80% 이상 단축했고, Tibero 기반 개발자가 기존 구조 그대로 자사 플랫폼에서 개발을 이어갈 수 있게 되어 학습 비용을 최소화했습니다.

기술 스택:

- 언어 : JavaScript(ES6+), TypeScript, Java
- 프레임워크 및 라이브러리 : ReactJS, React Flow, Jest
- 상태 관리 : MobX
- 협업 도구 : Figma, Slack

교육

숭실대학교

대학교(학사) | 컴퓨터학부
2016.03. ~ 2022.08. | 졸업

자격증

정보처리기사

한국산업인력공단
2022.01.